PRAKTIKUM MACHINE LEARNING II

ALGORITHM EVALUATION

Nama : Fellia Gessangie Yohanshah Puteri

NIM : 24060120130047

Lab : C1

---- SOAL 1 ----

1. import library pandas

Library ini digunakan untuk melakukan pengelolaan data yang berbentuk dataframe/tabel

```
import pandas
```

2. memasukkan dataset

Setelah import library, akan dimasukkan dataset yang akan digunakan pada praktikum ini. Adapun dataset yang digunakan menggunakan Iris Dataset yang diambil dari repository UCI Machine Learning

```
url = "http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.data"
names = ['sepal-length', 'sepal-width', 'petallength','petal-width','class']
datairis = pandas.read_csv(url, names=names)
```

3. import sklearn

Library adalah tool yang akan membantu dalam proses machine learning.

```
import sklearn
from sklearn import model_selection
```

4. validasi dataset

Pada bagian ini, dataset akan dibagi menjadi dua sebagai data train dan data test. Data train akan digunakan untuk melatih model dan sisanya akan masuk ke dalam data test untuk validasi model. Adapun pada praktikum ini, data akan dibagi sebesar 80% untuk data test dan 20% untuk data train.

```
array = datairis.values
X = array[:,0:4]
Y = array[:,4]
validation_size = 0.20
seed = 7
X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y, test_size=vali
```

5. import library algoritma klasifikasi

Pada bagian ini, dilakukan import library algoritma yang akan digunakan. Untuk proses klasifikasi digunakan 3 classifier dari sklearn, yaitu Decision Tree, Stochastic Gradient Descent, dan Gaussian Process.

```
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import SGDClassifier
from sklearn.gaussian_process import GaussianProcessClassifier
```

6. membuat array asosiatif

Array asosiatif ini digunakan untuk menyimpan algoritma di atas.

```
models = []

models.append(('Decision Tree',DecisionTreeClassifier()))
models.append(('Stochastic Gradient Descent',SGDClassifier()))
models.append(('Gaussian Process',GaussianProcessClassifier()))
```

7. membangun model

Pada bagian ini model mulai dibangun dengan menggunakan k-folds cross validation 10 kali lipat. Hasil pengujian model dapat dilihat dari output kode di bawah.

```
# evaluate each model in turn
results = []
names = []
for name, model in models:
  kfold = model_selection.KFold(n_splits=10, random_state=None)
  cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy'
  results.append(cv_results)
  names.append(name)
  msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
  print(msg)
```

```
Decision Tree: 0.975000 (0.038188)
Stochastic Gradient Descent: 0.866667 (0.124722)
Gaussian Process: 0.991667 (0.025000)
```

8. memilih model terbaik dan confusion matrix

Dapat dilihat bahwa algoritma Gaussian Process memiliki nilai accuracy terbesar sebesar 0.991667. Setelah itu, evaluasi algoritma dengan kode berikut.

```
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report

gp = GaussianProcessClassifier()
gp.fit(X_train, Y_train)
predictions = gp.predict(X_validation)
print('akurasi testing data \n', accuracy_score(Y_validation, predictions))
print('\n confusion matrix \n', confusion_matrix(Y_validation, predictions))
print('\n classification report \n', classification_report(Y_validation, predictions))
```

```
akurasi testing data
 0.8666666666666667

 confusion matrix
 [[ 7  0  0]
 [ 0 10  2]
 [ 0  2  9]]

 classification report
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00         7
Iris-versicolor       0.83      0.83      0.83        12
 Iris-virginica       0.82      0.82      0.82        11

       accuracy                           0.87        30
      macro avg       0.88      0.88      0.88        30
   weighted avg       0.87      0.87      0.87        30
```

---- SOAL 2 ----

note: import tidak lagi dilakukan karena ikut dalam soal 1

1. mount googledrive

Karena dataset pada repository berbentuk zip, saya akan memindahkan dataset tersebut ke dalam drive terlebih dahulu. Untuk mengambil dataset tersebut, dilakukan mount pada drive.

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

2. memasukkan dataset

Setelah mount drive, akan dimasukkan dataset yang akan digunakan pada praktikum ini. Adapun dataset yang digunakan menggunakan Raisin Dataset yang diambil dari repository UCI Machine Learning

```
dataraisin = pandas.read_excel('drive/My Drive/ml/raisindata.xlsx')
dataraisin.head(10)
```

	Area	MajorAxisLength	MinorAxisLength	Eccentricity	ConvexArea	Extent	Perimeter	Cl
0	87524	442.246011	253.291155	0.819738	90546	0.758651	1184.040	Kecin
1	75166	406.690687	243.032436	0.801805	78789	0.684130	1121.786	Kecin
2	90856	442.267048	266.328318	0.798354	93717	0.637613	1208.575	Kecin
3	45928	286.540559	208.760042	0.684989	47336	0.699599	844.162	Kecin
4	79408	352.190770	290.827533	0.564011	81463	0.792772	1073.251	Kecin
5	49242	318.125407	200.122120	0.777351	51368	0.658456	881.836	Kecin
6	42492	310.146072	176.131449	0.823099	43904	0.665894	823.796	Kecin
7	60952	332.455472	235.429835	0.706058	62329	0.743598	933.366	Kecin
8	42256	323.189607	172.575926	0.845499	44743	0.698031	849.728	Kecin
9	64380	366.964842	227.771615	0.784056	66125	0.664376	981.544	Kecin

3. validasi dataset

Pada bagian ini, dataset akan dibagi menjadi dua sebagai data train dan data test. Data train akan digunakan untuk melatih model dan sisanya akan masuk ke dalam data test untuk validasi model. Adapun pada praktikum ini, data akan dibagi sebesar 80% untuk data test dan 20% untuk data train.

```
array = dataraisin.values
X = array[:,0:7]
Y = array[:,7]
validation_size = 0.20
seed = 7
X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(X, Y, test_size=vali
```

4. import library algoritma klasifikasi

Pada bagian ini, dilakukan import library algoritma yang akan digunakan. Untuk proses klasifikasi digunakan classifier KNeighbors, GaussianNB, dan SVC.

```
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
```

5. membuat array asosiatif untuk algoritma

Array asosiatif ini digunakan untuk menyimpan algoritma di atas.

```
models2 = []
```

```
models2 = []

models2.append(('KNN',KNeighborsClassifier()))
models2.append(('NB',GaussianNB()))
models2.append(('SVM',SVC()))
```

6. evaluasi model

Pada bagian ini model mulai dibangun dengan menggunakan k-folds cross validation 10 kali lipat. Hasil pengujian model dapat dilihat dari output kode di bawah.

```
results2 = []
names2 = []

for name, model in models2:
  kfold = model_selection.KFold(n_splits=10, random_state=None)
  cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold, scoring='accuracy'
  results2.append(cv_results)
  names2.append(name)
  msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
  print(msg)
```

```
KNN: 0.826389 (0.047811)
NB: 0.820833 (0.043234)
SVM: 0.809722 (0.042605)
```

7. memilih model terbaik dan confusion matrix

Dapat dilihat bahwa algoritma KNN memiliki nilai accuracy terbesar sebesar 0.826389. Setelah itu, evaluasi algoritma dengan kode berikut.

```
knn = KNeighborsClassifier()
knn.fit(X_train, Y_train)
predictions = knn.predict(X_validation)
print('akurasi testing data \n', accuracy_score(Y_validation, predictions))
print('\n confusion matrix \n', confusion_matrix(Y_validation, predictions))
print('\n classification report \n', classification_report(Y_validation, predictions))
```

```
akurasi testing data
 0.8555555555555555

 confusion matrix
 [[84 17]
 [ 9 70]]

 classification report
               precision    recall  f1-score   support

        Besni       0.90      0.83      0.87       101
      Kecimen       0.80      0.89      0.84        79

     accuracy                           0.86       180
    macro avg       0.85      0.86      0.85       180
 weighted avg       0.86      0.86      0.86       180
```

Link Colab:

https://colab.research.google.com/drive/1gP6Rv52-_sCie0OR6fh0s3Tl6b-InJEW?usp=sharing